UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/804,843	03/19/2004	Richard D. Muratori	Intel-026PUS	7517

7590 09/06/2006

Daly, Crowley & Mofford, LLP
c/o PortfolioIP
P.O. Box 52050
Minneapolis, MN 55402

EXAMINER	
TRUONG, LOAN	

ART UNIT	PAPER NUMBER
2114	

DATE MAILED: 09/06/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/804,843	MURATORI ET AL.	
	Examiner	Art Unit	
	LOAN TRUONG	2114	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on 3/19/04.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) 1-19 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☒ Claim(s) 17-19 is/are allowed.

6)☒ Claim(s) 1-16 is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on 09 July 2004 is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Allowable Subject Matter

1. Claims 17- 19 are allowed.

The following is an examiner's statement of reasons for allowance:

The examiner deem claims 17-19 as novel when read as a whole for the limitations of multiple line cards interconnected by the switch fabric, individual ones of the line cards comprising a network processor including multi-threaded microengines configured to execute microcode, wherein the microcode includes instructions developed using a debugger tool that identified and enabled breakpoints to be set on multiple ones of the multi-threaded microengines, wherein the breakpoints correspond to a common line of source code in a common source code file.

Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

Claim Rejections - 35 USC § 112

The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

2. Claims 7-10 are rejected under 35 U.S.C. 112, second paragraph, the term "an article" is indefinite and fails to particularly point out and distinctly claim the subject

matter which applicant regards as the invention. Examiner suggests an article to be amended to "an article of manufacture".

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in a patent granted on an application for patent by another filed in the United States before the invention thereof by the applicant for patent, or on an international application by another who has fulfilled the requirements of paragraphs (1), (2), and (4) of section 371(c) of this title before the invention thereof by the applicant for patent.

The changes made to 35 U.S.C. 102(e) by the American Inventors Protection Act of 1999 (AIPA) and the Intellectual Property and High Technology Technical Amendments Act of 2002 do not apply when the reference is a U.S. patent resulting directly or indirectly from an international application filed before November 29, 2000. Therefore, the prior art date of the reference is determined under 35 U.S.C. 102(e) prior to the amendment by the AIPA (pre-AIPA 35 U.S.C. 102(e)).

3. Claims 1, 6-7 and 14-15 are rejected under 35 U.S.C. 102(e) as being anticipate by Bates et al. (US 6,681384).

In regard to claim 1, Bates et al. disclosed a method of debugging, comprising;

receiving a breakpoint selection for a program instruction associated with a first image file for a first processing engine (*dialog windows allows the user to request debug controller to insert a multi-threaded breakpoint into the program being debugged, fig. 13b, col. 9 lines 20-28)*;

identifying a source code file and a source code line in the source code file that generated the program instruction (*halt-at-statement allows user to specify the statement number where debug controller is to insert the break-point in the program being debugged, col. 9 lines 29-33)*;

identifying further processing engines having an image file containing a program instruction generated by the source code line in the source code file (*debug controller with break-point table where each record include address representing the physical storage location for the instruction in program at which break-point is activated, col. 10 lines 7-17)*; and

manipulating respective breakpoints for selected ones of the further processing engines based upon user selection (*dialog window contains condition which allow user to specify a condition that debug controller will check when determining when to activate a breakpoint, fig. 3, 350, 359, col. 9 lines 39-46)*, wherein the manipulated breakpoints correspond to program instructions generated by the source code line of the source code file (*user provides input that resumes execution of program 120, col. 7 lines 65-67).*

In regard to claim 6, Bates et al. disclosed the method according to claim 1, further including receiving a selection of breakpoint type (*type field, fig. 4, 430, col. 10 lines 29-31).*

In regard to claim 7, Bates et al. disclosed an article, comprising:

a storage medium having stored instructions (*main memory, fig. 1, 116)* thereon that when executed by a machine result in the following;

receiving a breakpoint selection for a program instruction associated with a first image file for a first processing engine (*dialog windows allows the user to request debug controller to insert a multi-threaded breakpoint into the program being debugged, fig. 13b, col. 9 lines 20-28)*;

identifying a source code file and a source code line in the source code file that generated the program instruction (*halt-at-statement allows user to specify the statement number where debug controller is to insert the break-point in the program being debugged, col. 9 lines 29-33)*;

identifying further processing engines having an image file containing a program instruction generated by the source code line in the source code file (*debug controller with break-point table where each record include address representing the physical storage location for the instruction in program at which break-point is activated, col. 10 lines 7-17)*; and

manipulating respective breakpoints for selected ones of the further processing engines based upon user selection (*dialog window contains condition which allow user to specify a condition that debug controller will check when determining when to activate a breakpoint, fig. 3, 350, 359, col. 9 lines 39-46)*, wherein the manipulated breakpoints correspond to program instructions generated by the source code line of the source code file (*user provides input that resumes execution of program 120, col. 7 lines 65-67)*.

In regard to claim 14, Bates et al. disclosed a debugger tool system, comprising: a processor (*processor, fig. 1, 112)*; and

memory (*main memory, fig. 1, 116)* coupled to the processor , wherein the processor and memory combine to execute instructions that result in the following:

receiving a breakpoint selection for a program instruction associated with a first image file for a first processing engine (*dialog windows allows the user to request debug controller to insert a multi-threaded breakpoint into the program being debugged, fig. 13b, col. 9 lines 20-28)*;

identifying a source code file and a source code line in the source code file that generated the program instruction (*halt-at-statement allows user to specify the statement number where debug controller is to insert the break-point in the program being debugged, col. 9 lines 29-33)*;

identifying further processing engines having an image file containing a program instruction generated by the source code line in the source code file (*debug controller with break-point table where each record include address representing the physical storage location for the instruction in program at which break-point is activated, col. 10 lines 7-17)*; and

manipulating respective breakpoints for selected ones of the further processing engines based upon user selection (*dialog window contains condition which allow user to specify a condition that debug controller will check when determining when to activate a breakpoint, fig. 3, 350, 359, col. 9 lines 39-46)*, wherein the manipulated breakpoints correspond to program instructions generated by the source code line of the source code file (*user provides input that resumes execution of program 120, col. 7 lines 65-67).*

In regard to claim 15, Bates et al. disclosed the system according to claim 14, wherein manipulating the respective breakpoints includes one or more of inserting breakpoints (*Set breakpoint, fig. 3a, 301)*, removing breakpoint (*Delete breakpoint, fig. 3a, 304)*, enabling breakpoints and disabling breakpoints (*enabling and disabling breakpoint are accomplished by satisfying conditions that user input through the dialog windows, col. 9 lines 29-67 and col. 10 lines 1-6).*

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

The factual inquiries set forth in *Graham* **v.** *John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

4. Claims 2-5, 8-13, 16 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bates et al. (US 6,681384) in further view of Hunter et al. (US 2002/0100024).

In regard to claim 2, Bates et al. does not teach the method according to claim 1, further including displaying a first screen that includes a mechanism to list the further processing engines.

Hunter et al. teach a method of shared software breakpoints in a shared memory system having a setup utility interface where the third level of hierarchy lists the processors on the target board (*fig. 3, 300, paragraph 0037).*

It would have been obvious to modify the method of Bates et al. by adding Hunter et al. method of shared software breakpoints. A person of ordinary skill in the art at the time of applicant's invention would have been motivated to make the modification because it would help in the debug and software development of system-on-a-chip architectures for embedded system (*paragraph 0004).*

In regard to claim 3, Bates et al. does not teach the method according to claim 2, further including displaying a second screen to enable a user to select the further ones of the processing engines.

Hunter et al. teach a method of shared software breakpoints in a shared memory system to active a debug window for selected processor from the Open Dialog of parallel debug manager (*fig. 4, 402, paragraph 0071).*

Refer to claim 2 for motivational statement.

In regard to claim 4, Bates et al. teach the method according to claim 3, further including displaying the second screen (*dialog window, fig. 3a, 300)* to enable the user to manipulate the breakpoints as one or more of inserting the breakpoint (*Set breakpoint,*

fig. 3a, 301), removing the breakpoint (*Delete breakpoint, fig. 3a, 304)*, enabling the breakpoint and disabling the breakpoint (*enabling and disabling breakpoint are accomplished by satisfying conditions that user input through the dialog windows, col. 9 lines 29-67 and col. 10 lines 1-6).*

Bates et al. does not teach the method of selected ones of the further processing engines.

Hunter et al. teach a method of shared software breakpoints in a shared memory system to active a debug window for selected processor from the Open Dialog of parallel debug manager (*fig. 4, 402, paragraph 0071).*

Refer to claim 2 for motivational statement.

In regard to claim 5, Bates et al. does not teach the method according to claim 1, further including identifying the further processing engines by searching image files for the further processing engines to identify source code files for the image files containing the source code line.

Hunter et al. teach the method of shared software breakpoints in a shared memory system by having the bus manager searches all memory maps associated with the debug session to locate processors that share the memory location (*fig. 11, 1106, 1102, 1104, 510, paragraph 0076*).

Refer to claim 2 for motivational statement.

In regard to claim 8, Bates et al. does not teach disclosed the article according to claim 7, further including stored instructions to enable displaying a first screen that includes a mechanism to enable a user to see the further processing engines.

Hunter et al. teach a method of shared software breakpoints in a shared memory system having a setup utility interface where the third level of hierarchy lists the processors on the target board (*fig. 3, 300, paragraph 0037).*

Refer to claim 2 for motivational statement.

In regard to claim 9, Bates et al. does not teach the article according to claim 8, further including stored instructions to enable displaying a second screen to enable a user to select the further ones of the processing engines.

Hunter et al. teach a method of shared software breakpoints in a shared memory system to active a debug window for selected processor from the Open Dialog of parallel debug manager (*fig. 4, 402, paragraph 0071).*

Refer to claim 2 for motivational statement.

In regard to claim 10, Bates et al. teach the article according to claim 9, further including stored instructions to enable displaying the second screen (*Dialog window, fig. 3a, 300)* to enable the user to select one or more of inserting the breakpoint (*Set breakpoint, fig. 3a, 301)*, removing the breakpoint (*Delete breakpoint, fig. 3a, 304)*, enabling the breakpoint and disabling the breakpoint (*enabling and disabling breakpoint are accomplished by satisfying conditions that user input through the dialog windows, col. 9 lines 29-67 and col. 10 lines 1-6).*

Bates et al. does not teach the article further including selected ones of the further processing engines

Hunter et al. teach a method of shared software breakpoints in a shared memory system to active a debug window for selected processor from the Open Dialog of parallel debug manager *(fig. 4, 402, paragraph 0071).*

Refer to claim 2 for motivational statement.

In regard to claim 11, Bates et al. disclosed a graphical user interface, comprising:

a first window to show microcode instructions *(code, fig. 3a)* associated with a first image file *(statement, fig. 3a)* for a first processing engine *(threads, fig. 3a, 325, 327, 329)* in a processor simulator *(debugger, fig. 3a, 300)* including a breakpoint for a first one of the microcode instructions *(code, fig. 3a)* generated by a source code line in a source code file *(statement number of computer program, fig. 3a, 330-332, col. 8 lines 28-30)*;

a first menu to show user options including a first option to set breakpoints *(Debugger, set breakpoint, fig. 3a, 301)*; and

a second window *(Dialog window, fig. 3a, fig. 3b, 300, 350)* to display further processing engines *(threads, fig. 3a, 325, 327, 329, col. 8 lines 30-33)* having respective image files containing microcode instructions generated from the source code line *(statement number of computer program, fig. 3a, 330-332, col. 8 lines 28-30)* and to enable the user to manipulate respective breakpoints *(dialog window contains condition which allow user to specify a condition that debug controller will check when*

determining when to activate a breakpoint, fig. 3, 350, 359, col. 9 lines 39-46) for the further processing engines.

Bates et al. does not teach the graphical user interface comprising of the multiple processing engines.

Hunter et al. teach a method of shared software breakpoints in a shared memory system to active a debug window for selected processor from the Open Dialog of parallel debug manager *(fig. 4, 402, paragraph 0071).*

Refer to claim 2 for motivational statement.

In regard to claim 12, Bates et al. disclosed the graphical user interface according to claim 11, wherein the second window *(Dialog window, fig. 3b, 350)* includes icons to manipulate the respective breakpoints by one or more of inserting breakpoints *(Set breakpoint, fig. 3a, 301)*, removing breakpoint *(Delete breakpoint, fig. 3a, 304)*, enabling breakpoints and disabling breakpoints *(enabling and disabling breakpoint are accomplished by satisfying conditions that user input through the dialog windows, col. 9 lines 29-67 and col. 10 lines 1-6).*

In regard to claim 13, Bates et al. disclosed the graphical user interface according to claim 12, wherein the second window includes a display of the respective image files *(statement number of computer program, fig. 3a, 330-332, col. 8 lines 28-30).*

In regard to claim 16, Bates et al. does not teach the system according to claim 14, further including displaying a list of the identified further processing engines.

Hunter et al. teach a method of shared software breakpoints in a shared memory system having a setup utility interface where the third level of hierarchy lists the processors on the target board *(fig. 3, 300, paragraph 0037).*

Refer to claim 2 for motivational statement.

Conclusion

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. See PTO 892.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Loan Truong whose telephone number is (571) 272-2572. The examiner can normally be reached on M-F from 8am-4pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Scott Baderman can be reached on (571) 272-3644. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Loan Truong

Patent Examiner
AU 2114

SCOTT BADERMAN
SUPERVISORY PATENT EXAMINER